

Via E-File

PATENT APPLICATION

Docket: 13768.132

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE  
BEFORE THE BOARD OF PATENT APPEALS AND  
INTERFERENCES

In re application of		)
		)
	Damon Barry et al.	)
		)
Serial No.:	09/607,397	) Art Unit
		) 2192
Filed:	June 30, 2000	)
		)
Conf. No.:	9886	)
		)
For:	SYSTEMS AND METHODS FOR GATHERING	)
	ORGANIZING AND EXECUTING TEXT CASES	)
		)
Examiner:	Eric B. Kiss	)
		)
Customer No.:	47973	)

REPLY BRIEF OF APPELLANT

VIA eFILE Appeal Brief - Patents  
Commissioner for Patents  
P.O. Box 1450  
Alexandria, VA 22313-1450

Dear Sir:

On June 12, 2007, the Examiner mailed its Examiner's Answer to Appellant's timely-filed Appeal Brief. This Reply Brief is being filed under the provisions of 37 C.F.R. § 41.41.

This brief is being filed on Friday, August 10, 2007, and is therefore timely under 37 C.F.R. §§ 41.41 and 1.7.

## ARGUMENT

### I. Introduction

Appealed claims 1, 2, 4, 7, 10-17, and 20-28 stand rejected under 35 U.S.C. § 102(b), and appealed claim 3 stands rejected under 35 U.S.C. § 103(a). Each of the appealed claims requires the controller to organize a “test case hierarchy in which one or more test cases comprise a test suite, and in which one or more test suites comprise a test module[.]” Each of the appealed claims further requires the controller to “scan and discover” the test cases. In making his § 102(b) and § 103(a) rejections, the Examiner cites the TETware Release 3.3 software product released September 18, 1998 by The Open Group, as evidenced by: “TETware User Guide, Revision 1.2”, “Release Notes for TETware Release 3.3” and “TETware Programmers Guide, Revision 1.2.” Specifically, the Examiner argues that the TETware documentation teaches the claimed test case hierarchy and controller that scans and discovers the test cases. The primary issues before the Board are whether the TETware documentation teaches 1) the claimed “test case hierarchy” organized by a controller, and 2) the claimed test case discovery by the controller. Neither of these claimed features are found in the cited TETware documentation, as explained below.

### II. The Examiner Concedes that the “Test Scenario” of the TETware Documentation Does Not Teach the Claimed “Test Hierarchy”

At pages ten and eleven of the Appeal Brief, the Applicant explains why the “test scenario” of the TETware documentation can not be construed as a position in the claimed “test

hierarchy.” Specifically, the TETware “test scenario” is a separate set of instruction for using the TETware “test suite” and “test case,” and therefore shares no hierarchical position with the “test suite” and “test case.” The Examiner does not challenge this assertion by the Applicant, and therefore tacitly concedes that the TETware “test scenario” does not teach any part of the claimed “test hierarchy.”

III. The TETware “Test Suite Root Directory” and “Tet Root Directory” Are Not Organized by the Controller, and Therefore Can Not Be the Claimed “Test Hierarchy”

The Examiner argues in the Examiner’s Answer that the TETware “test suite root directories” and “tet root” directory teach the claimed “test module.” Examiner’s Answer, at p. 11. The Examiner’s argument confuses a hierarchical directory of repositories of test suites and test cases *from which test suites are pulled for execution* with the claimed “test case hierarchy” *organized by the controller*.

The claims explicitly recite “a harness client comprising a set of instructions that . . . (ii) executes a connector to scan for and discover the one or more available test cases that are stored in the one or more program modules and to organize the one or more available test cases into a test case hierarchy[.]” *See, e.g.*, claim 1. Thus, according to the claims, the test cases are “stored” in repositories called “program modules.” Importantly, these “program modules” where the test cases are stored are not part of the “test case hierarchy” that is “organized” by the controller.

The Examiner cites to section 5.2.6, titled “TETware Directory Layout,” of the TETware User Guide (“TET\_UG”) as teaching the claimed “test case hierarchy.” Examiner’s Answer, at

p. 11. Section 5.2.6, however, teaches that the directories are not organized by the controller, and therefore do not teach the claimed “test case hierarchy.” The Examiner argues that “TETware uses a Test Case Controller (tcc) to process test suite scenarios.” Examiner’s Answer, at p. 13. In other words, the Examiner argues that the tcc is the claimed “controller.” Section 5.2.6, however, teaches that “[b]y default, tcc *looks for* test cases below the “test suite root directory.” (Emphasis added). Because the tcc (i.e., the Examiner’s argued “controller”) *looks for* test cases in the “test suite root directory,” the “test suite root directory” can not be part of the claimed “test case hierarchy” that is *organized* by the tcc.<sup>1</sup> In other words, the “test suite root directory” and the “tet root directory” are simply repositories where the tcc goes to *find* test cases which, according to the claims, are subsequently organized by the controller. Consequently, neither the “test suite root directory” nor the “tet root directory” teach any part of the claimed “test case hierarchy” that is “organized” by the “controller.”

The Examiner also admits that prior to invoking the tcc, “the user is required to ensure that the value of the TET\_ROOT environment variable points to the “tet root directory” on the local system.” Examiner’s Answer, at p. 13. In other words, the user must direct the tcc to the “tet root directory” *before invoking the tcc*. See TET\_UGG, at § 6.2.2. It would not be possible to point to the tcc to the “tet root directory” *before invoking the tcc* if it is the tcc that organizes the “tet root directory”, as argued by the Examiner. Clearly, the “tet root directory” exists prior

---

<sup>1</sup> The Examiner concedes that “the “test suite root directory” is usually located immediately below the ‘tet root’ directory.” Examiner’s Answer, at p. 11. Thus, because the “test suite root directory” is not a position in the claimed “test case hierarchy,” it follows that the “tet root directory” (a subset of the “test suite root directory”) can not be a position in the claimed “test case hierarchy.”

to invoking to the tcc, and therefore the tcc does not “organize” the “tet root directory” into a “test case hierarchy” as required by the claims. Thus, the TETware documentation does not teach the claimed “test case hierarchy,” and the Examiner’s rejections based thereon should respectfully be removed.

The Examiner also argues that Figures 17 and 3 of the TETware Programmers Guide (“TET\_PG”) teach an example of the “test suite root directory.” Examiner’s Answer, at p. 12. Having established above that the TETware “test suite root directory” and the “tet root directory” are not organized by a controller, Examiner’s argument regarding Figures 17 and 3 of the TET\_PG are misplaced for the same reasons discussed above.

#### IV. The TETware Documentation Does Not Teach the Claimed Test Case Discovery

The claims require that the harness client “receives user input specifying one or more filenames corresponding to the one or more program modules” and “executes a connector to scan for and discover the one or more available test cases that are stored in the one or more program modules[.]” *See, e.g.*, claim 1. Importantly, the user specifies merely the “program modules,” and the connector “scans for and discovers” the available test cases in those program modules. These limitations are not taught in the TETware documentation as argued by the Examiner.

The Examiner argues that the TET\_PG teaches the claimed test case discovery at sections 6.2.2, 6.2.3, and 5.3.2. The Examiner’s arguments regarding section 5.2.3 was addressed in the Appeal Brief, and will not be reiterated here. The Examiner’s reliance on sections 6.2.2 and 6.2.3 of the TET\_PG as teaching the claimed test case discovery is misplaced.

Section 6.2.2 of the TET\_UG is consists of one sentence that states, “Before you invoke tcc, you must ensure that the value of your TET\_ROOT environment variable points to the “tet root directory” on the local system.” (Emphasis in original). Nothing in this section discusses whether the tcc “scan[s] for and discover[s] the one or more available test cases[.]”

Section 6.2.3 of the TET\_UG teaches that when invoked, the tcc “processes each test case in the specified scenario. (If no scenario is specified, then tcc processes each test case in the scenario named all).” However, the “processing” of test cases is different from “scanning for and discovering” test cases. Indeed, claim 1 specifically recites, “a harness client comprising a set of instructions that (i) receives user input specifying one or more filenames corresponding to the one or more program modules, (ii) executes a connector to scan for and discover the one or more available test cases that are stored in the one or more program modules . . . , and (iii) *receives user input indicating which of the* one or more available *test cases* in the test case hierarchy *are selected to be executed* on the computer program[.]” (Emphasis added). Thus, the execution or “processing” of the test cases is a step different from “scanning for and discovering” test cases. Consequently, section 6.2.3 of the TET\_UG teaching how the tcc “*processes* each test case” can not be interpreted to also teach the limitation of “scanning for and discovering” the test cases. The Examiner has failed to point to any portion of the cited TETware documentation teaching the claimed test case discovery. Consequently, the Examiner’s rejection can not stand and should respectfully be removed.

### CONCLUSION

For the reasons set forth above, and the additional reasons set forth in the Appeal Brief, Appellant respectfully requests the Board to overturn the Examiner's rejections of the appealed claims 1-4, 7, 10-17, and 20-28 and to allow those claims in their present form.

Dated August 10, 2007.

Respectfully submitted,

/ADRIAN J. LEE/

ADRIAN J. LEE  
Attorney for Applicant  
Registration No. 42,785  
Customer No. 047973

CN0000000016V001